

Implementación en tiempo real para el seguimiento en secuencias de vídeo

Jesús Eduardo Alcaraz Chávez

Universidad Michoacana de San Nicolás de Hidalgo,
Facultad de Ingeniería Eléctrica División de Estudios de Posgrado
likcos@gmail.com

Félix Calderón Solorio.

Universidad Michoacana de San Nicolás de Hidalgo,
Facultad de Ingeniería Eléctrica División de Estudios de Posgrado
calderon@umich.mx

Resumen: En el presente trabajo se implementa un seguidor en secuencias de vídeo. La implementación trabaja en tiempo real. El seguidor, ésta basado en el algoritmo de SURF (Speed Up Robust Feature), debido a la robustez que presenta a los cambios de luminosidad y a las posibles transformaciones geométricas de una imagen, como lo son: el escalamiento, cizallamiento, rotación, translación. El objetivo central de este trabajo es obtener un seguidor en tiempo real, que no necesite del objeto de interés un aprendizaje o estudio previo y por ende se pueda generalizar lo que se quiere seguir. Es decir, poder tomar cualquier objeto de una escena cualquiera. El proceso se inicia una vez se obtiene la muestra del objeto a seguir. Posteriormente se generan los puntos de interés mediante el algoritmo de SURF, tanto del objeto de muestras como de la secuencia de video. Así mismo mediante vecinos cercanos se busca la correspondencia entre los puntos generados de la muestra y de la secuencia de videos que mejor correspondan, finalmente mediante el algoritmo de RANSAC se toman un conjunto de puntos, para buscar la homografía de la figura que mejor se adapte al objeto de interés, finalmente se demarca el objeto de interés con los mejores parámetros homográficos encontrados, todo esto en tiempo real.

Palabras clave: Seguimiento, SURF, transformaciones geométricas, RANSAC, vecinos próximos, puntos claves.

Implementation in real time tracking in video sequences

Abstract: In this paper a tracker is implemented in video sequences. The implementation works in real time. The tracker is based on the algorithm of SURF (Speed Up Robust Feature) due to the robustness Presenting the light changes and the possible geometric transformations of an image, such as : scaling , shearing , rotation, translation. The main objective of this work is to obtain a real-time tracker that does not need the object of interest prior learning or study and therefore can generalize what you want to follow. I mean, to take anything of any scene . The process begins once the sample is obtained subject to follow. Then the points of interest are generated by the SURF algorithm, both the object and the sample video sequence . Also by neighbors correspondence between sample points generated and the sequence of videos that best match finally by RANSAC algorithm seeks a set of points are taken to find the homography of the figure that best suits the object of interest, finally the object of interest is demarcated with the best parameters found homographs , all in real time.

Keywords: Tracking, SURF, geometric transformations, RANSAC, nearest neighbors, key point

1. Introducción

En el presente trabajo se implementa un seguidor en secuencias de vídeo. La implementación trabaja en tiempo real. El seguidor, ésta basado en el algoritmo de SURF (Speed Up Robust Feature), debido a la robustez que presenta a los cambios de luminosidad y a las posibles transformaciones geométricas de una imagen, como lo son: el escalamiento, cizallamiento, rotación, translación. El objetivo central de este trabajo es obtener un seguidor en tiempo real, que no necesite del objeto de interés un aprendizaje o estudio previo y por ende se pueda generalizar lo que se quiere seguir. Es decir, poder tomar cualquier objeto de una escena cualquiera. El proceso se inicia una vez se obtiene la muestra del objeto a seguir. Posteriormente se generan los puntos de interés mediante el algoritmo de SURF, tanto del objeto de muestras como de la secuencia de video. Así mismo mediante vecinos cercanos se busca la correspondencia entre los puntos generados de la muestra y de la secuencia de

videos que mejor correspondan, finalmente mediante el algoritmo de RANSAC se toman un conjunto de puntos, para buscar la homografía de la figura que mejor se adapte al objeto de interés, finalmente se demarca el objeto de interés con los mejores parámetros holográficos encontrados, todo esto en tiempo real.

2. Metodología

Kalal propone en [Kalal et al., 2010] la adquisición de un objeto de interés mediante un solo click en un cuadro de la secuencia de vídeo, una vez adquirido el objeto de interés, se inicia el seguimiento generando la trayectoria del objeto. Puesto que el enfoque de Kalal, [Kalal et al., 2010] que es una aportación nueva e innovadora al aprendizaje automático y de tiempo real, el cual a partir de la generación de la trayectoria se va adaptando a las modificaciones del objeto. Debido a que obtiene las características a partir de los pequeños cambios del objeto, para posteriormente ser guardados en una memoria temporal, de esta forma, cuando el objeto se pierde, puede ser detectado nuevamente por esta memoria, clasificador, que se va generando en el transcurso del seguimiento. Por ende, es notable sin duda la idea, dado que de un solo ejemplo, se puede hacer el seguimiento y la detección por la memoria generada, es decir, mientras se hace el seguimiento, se sabe que es el objeto de interés, de esta forma, es posible discriminar objetos que aparezcan en la escena. Luego entonces, basándose en la idea de Kalal, donde se puede generalizar lo que se requiere seguir, surge la idea de implementar un seguidor con características similares. Basando el trabajo en algoritmos generadores de características, robustos a la invariancia de rotación, traslación, cambios en la iluminación. La idea se complementa al observar el aprendizaje del ser humano, que basa la discriminación de objetos y su clasificación en características para poder hacer una identificación adecuada de cualquier objeto de la vida cotidiana. Debido a ello, es que se implementó el algoritmo de SURF [Bay et al., 2008], dado que a partir de una sola muestra se pueden obtener puntos verdaderamente característicos, con los que es posible hacer la detección y el seguimiento con

una sola muestra del objeto de interés, por tal razón, se aborda en el siguiente apartado.

3. SURF (Speeded Up RobustFeatures)

El descriptor SURF, cuyo acrónimo hace referencia al título, Speed-Up Robust Feature, fue desarrollado por Herbert Bay[Bay et al., 2008] como un detector de puntos de interés y descriptor robusto. El descriptor SURF guarda cierto parecido con la filosofía del descriptor SIFT [Lowe, 2004] si bien presenta cambios notables, los autores afirman que este detector y descriptor presenta 2 mejoras resumidas en los siguientes conceptos:

- Velocidad de cálculo considerablemente superior sin ocasionar pérdida del rendimiento.
- Mayor robustez ante posibles transformaciones de la imagen.

El descriptor SURF hace uso de la matriz Hessiana, más concretamente, del valor del determinante de la matriz, para la localización y la escala de los puntos. El motivo para la utilización de la matriz Hessiana es que se generan puntos como gota y se hace uso de la curvatura principal, dando como resultado, la robustez que se menciona ante las diferentes transformaciones geométricas. Así mismo lo realmente novedoso del detector incluido en el descriptor SURF respecto de otros detectores es que no utiliza diferentes medidas para el cálculo de la posición y la escala de los puntos de interés individualmente, sino que utiliza el valor del determinante de la matriz Hessiana en ambos casos. Por lo tanto dado un punto $p=[x,y]^T$ de la imagen I , la matriz Hessiana $H(p,\sigma)$ del punto I perteneciente a la escala σ se define como:

$$H(p,\sigma) = \begin{bmatrix} D_{xx}(p,\sigma) & D_{xy}(p,\sigma) \\ D_{xy}(p,\sigma) & D_{yy}(p,\sigma) \end{bmatrix}. \quad (1)$$

Donde $D_{xx}(p, \sigma)$ representa la convolución de la derivada de segundo orden de la Gaussiana $\partial^2 / \partial x^2 g(\sigma)$ con la imagen I en el punto p . De manera análoga ocurre con los términos $D_{xy}(p, \sigma)$, $D_{yy}(p, \sigma)$ de la matriz. A pesar de que los filtros gaussianos son óptimos para el análisis del espacio-escala [Lindeberg, 1990], se ha implementado una alternativa a los filtros gaussianos en el detector SURF debido a una serie de limitaciones de estos filtros como la necesidad de ser discretizados, la falta de prevención total del indeseado efecto aliasing. Por ende, se utilizan los filtros tipo caja (de sus siglas en inglés box-filters) [Simard et al., 1998]. Estos filtros aproximan las derivadas parciales de segundo orden de las Gaussianas y pueden ser evaluados de manera muy rápida usando imágenes integrales, cuya definición se encuentra ampliamente detallada en [Derpanis et al., 2007], [Viola y Jones, 2004].

4. Imágenes integrales

Considerando I' como una imagen, en tono de grises (tomando valores en el rango de 0 a 255), lo cual sería una forma de aplicar robustez al cambio de color (tales como una corrección de balance de blancos). La imagen integral se define de la siguiente manera.

$$I_{\Sigma}(x,y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I'(i,j). \quad (2)$$

En la Figura 1 se ilustra el valor de la imagen integral en el punto $[x,y]^T$, donde se muestra la suma de todas las intensidades. Una imagen integral puede ser creada de manera recursiva, comienza en la esquina superior izquierda y trabaja por una fila a la vez. La suma de las intensidades en el cuadro dorado es la siguiente donde u y v son coordenadas del otro punto que forma el rectángulo:

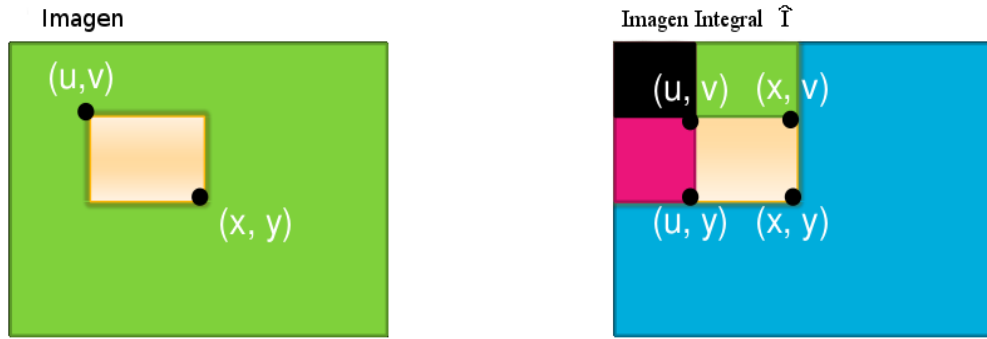


Figura 1: Ejemplo de Imagen Integral.

El cálculo de tal imagen integral I_c permite convolucionar la imagen I' con una caja en tres operaciones y cuatro accesos a memoria usando la siguiente fórmula.

$$c * I'(x, y) = I_c(x, y) - I_c(x, v) - I_c(u, y) + I_c(u, v). \quad (2)$$

Con esto se acelera el proceso de la generación de puntos característicos y da la posibilidad de hacer el seguimiento a tiempo real.

5. Muestreo del espacio-escala

El espacio-escala en el enfoque en que SURF está basado en un muestreo regular de parámetros de escala L . La representación de escala es dividida en octavas: una nueva octava corresponde con el doble del tamaño del kernel (lo que significa que las escalas son una serie geométrica). Cada octava es dividida en diferentes niveles (o intervalos), cada una corresponde al incremento del tamaño del filtro implicado. En SURF, se define como el índice de octava e i el índice del intervalo, el muestreo de escala se obtiene con la siguiente relación:

$$\sigma = \frac{1.2}{3} (2^s \times i + 1) = \frac{1.2}{3} \times L = 0.4L \quad (3)$$

donde $0.4L$ es la escala que se utiliza para todos los cálculos de puntos clave y se corresponde con el parámetro de detección de escala, el análogo del parámetro de la longitud de la caja. Una fórmula más precisa debería haber sido utilizada, desde.

$$\sigma(B_L) \approx \frac{1}{\sqrt{12}} \approx 0.29L \quad (4)$$

Debido a la utilización de filtros de caja e imágenes integrales no se tiene que aplicar iterativamente el mismo filtro a la salida de una capa previamente filtrada, en su lugar se aplican filtros de caja de tamaños previamente expuestos exactamente a la misma velocidad, directamente en la imagen original e incluso en paralelo. Por lo tanto, el espacio-escala se analiza incrementando el tamaño del filtro en lugar de reducir iterativamente el tamaño de la imagen como se muestra en la Figura 2.

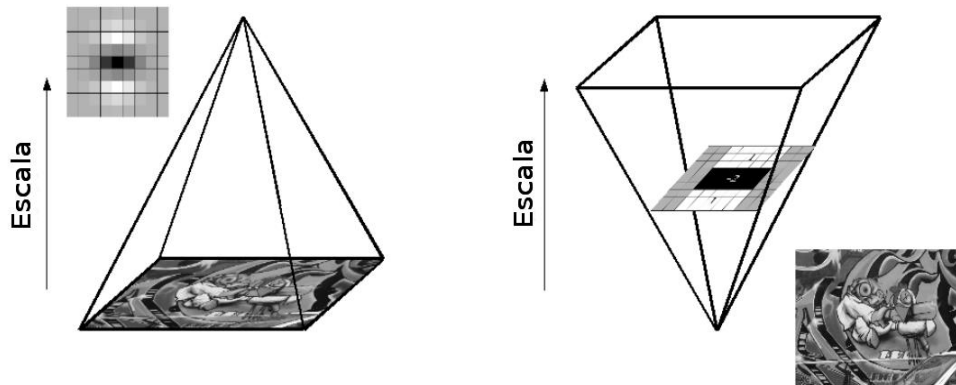


Figura 2: En lugar de reducir iterativamente el tamaño de la imagen (a la izquierda), el uso de imágenes integrales permite el aumento de escala del filtro con un costo constante.

6. Selección de características

Los puntos de interés son puntos en el espacio-escala que corresponden a un máximo local citado en el operador del determinante del Hessiano, aplicado a la representación del espacio-escala en la imagen. Estos puntos son seleccionados considerando un vecindario $3 \times 3 \times 3$, realizando así una comparación exhaustiva de todos los puntos del espacio-escala con sus 26 vecinos más cercanos. Con el fin de obtener una representación compacta de la imagen y también para hacer frente al ruido, el algoritmo selecciona la característica más sobre (i.e) de un conjunto de máximos locales, esto es logrado usando un umbral tH en la respuesta del operador del determinante del Hessiano para cada uno de los puntos de interés candidatos $[x,y]^T$ en la escala L , como se puede ver en la Figura 3.

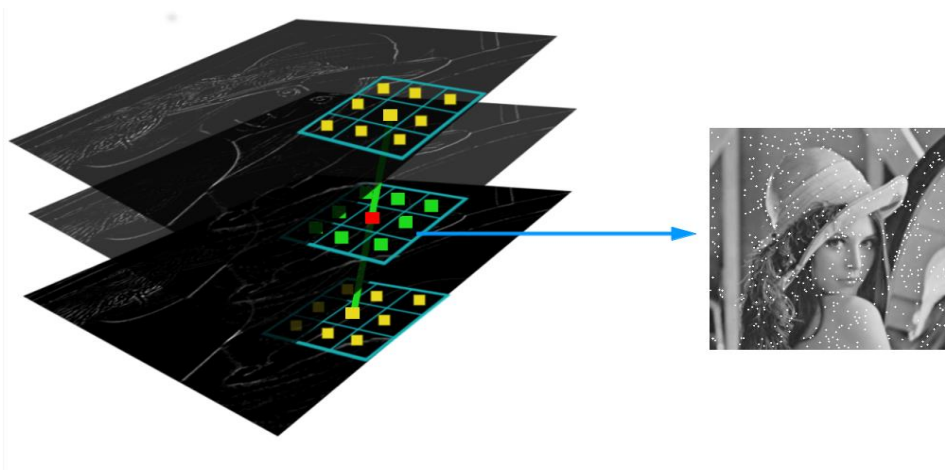


Figura 3: En rojo: Píxel en estudio. En verde: Vecinos en escala actual. En amarillo: Vecinos de escala anterior y posterior.

7. Descripción local

De la etapa anterior, se obtiene un conjunto de puntos de interés N en el espacio-escala $M_i: (x_i, y_i, L_i), \dots, N$ lo cual permite codificar características invariantes a escala en la imagen. Sin embargo antes de construir el descriptor local, primero se tiene que definir para cada punto de interés una orientación dominante siguiendo el procedimiento que se detalla en los párrafos subsecuentes, a fin de lograr invariancia total a la rotación.

8. Creación del descriptor

Es en esta última etapa del proceso donde se concreta la creación del descriptor SURF. Se construye como primer paso una región cuadrada de tamaño $20L_i$ alrededor del punto de interés y orientada en relación a la orientación calculada en la etapa anterior. Esta región es a su vez dividida en 4×4 sub-regiones dentro de cada una de las cuales se calculan las respuestas de Haar de puntos con una separación de muestreo de 5×5 en ambas direcciones. Por simplicidad, se consideran dx y dy las respuestas de Haar en las direcciones horizontal y vertical, respectivamente, relativas a la orientación del punto de interés. En la Figura 4 están representadas tanto las respuestas de Haar en cada una de las sub-regiones como las componentes dx y dy uno de los vectores. Para dotar a las respuestas dx y dy de una mayor robustez ante deformaciones geométricas y errores de posición, éstas son ponderadas por una Gaussiana de valor $\sigma = 3,3\sigma_i$ centrada en el punto de interés. En cada una de las sub-regiones se suman las respuestas dx y dy obteniendo así un valor de dx y dy representativo por cada una de las sub-regiones. Al mismo tiempo se realiza la suma de los valores absolutos de las respuestas $|dx|$ y $|dy|$ en cada una de las sub-regiones, obteniendo de esta manera, información de la polaridad sobre los cambios de intensidad.

$$v = \left(\sum_{i=1}^{32} dx, \sum_{i=1}^{32} dy, \sum_{i=1}^{64} |dx|, \sum_{i=1}^{64} |dy| \right) \quad (5)$$

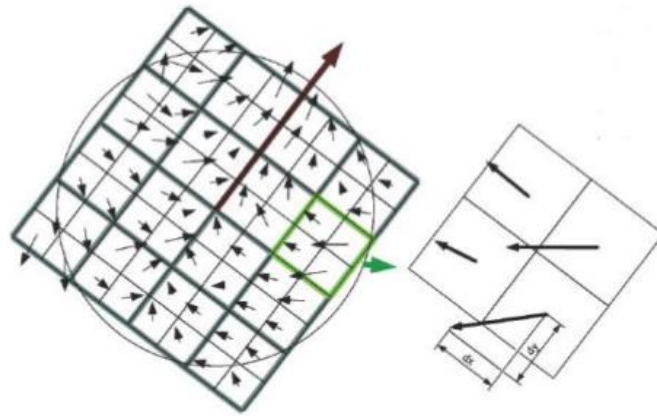


Figura 4: Respuestas de Haar en las sub-regiones alrededor del punto de interés

9. Correspondencias de características

De los pasos anteriores, dos imágenes correspondientes pueden ser representadas por dos conjuntos de puntos de interés con su correspondiente descriptor SURF. Recordando que el descriptor SURF es un vector con una dimensionalidad de 64 compartimientos, con norma unitaria Euclidiana. El paso de la correspondencia se realiza hasta este punto, comparando exhaustivamente los vectores, combinado con la distancia del radio del vecindario más próximo (NN-DR). La distancia Euclidiana entre el descriptor de la primera imagen y cualquier descriptor de la segunda imagen son calculados, solo los más cercanos y los segundos vecinos más cercanos son considerados. Si el radio de estas dos distancias es menor a 0.8 la correspondencia entre el descriptor consultado y su candidato es validado. Tal umbral permite descartar numerosas discordancias, un ejemplo de lo que se menciona en la

correspondencia de características puede ser visto en la Figura 5. Una vez que se ha calculado los descriptores

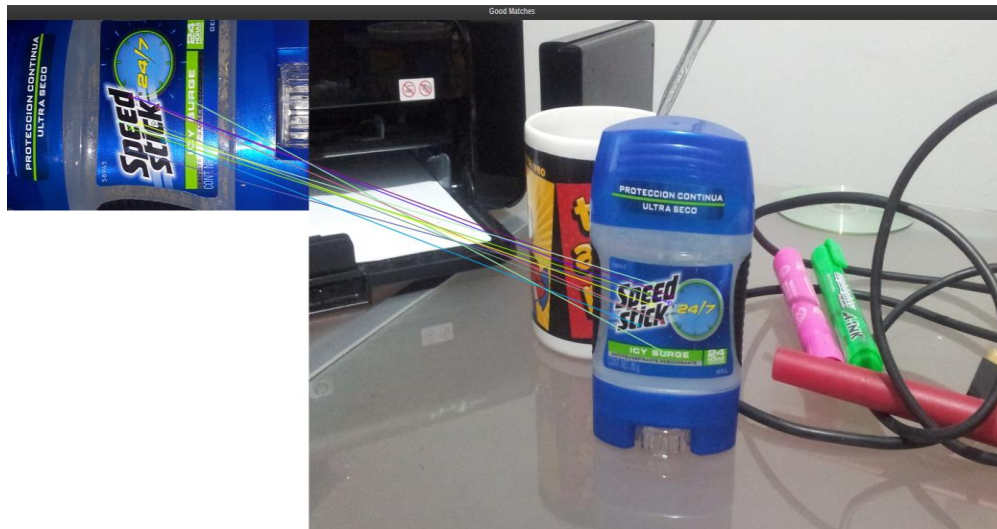


Figura 5: Correspondencia de Puntos de Interés

Sólo queda realizar la correspondencia entre los puntos de interés, ya sea de dos imágenes o de un conjunto de imágenes.(e.q.). Para ello Lowe [Lowe, 2004] propone el método de vecinos más próximos. Este método está basado en la distancia Euclidiana entre vectores. El algoritmo consiste en calcular los descriptores de una imagen A, en el cual se quiere encontrar un objeto que está definido en una imagen B. Cuando ya se tienen los vectores de los descriptores de todos los puntos de interés, por cada punto de interés de la imagen A se calcula una distancia que existe entre un vector descriptor y todos los demás vectores de los puntos característicos de la imagen B. Una vez que se tiene calculadas todas las distancias por cada punto de interés de la imagen A, se calcula la relación entre las dos distancias menores. Si esta relación es menor que el valor umbral, existe una correspondencia entre el punto de interés de A y el punto de interés de B más cercano. Para calcular el grado de similitud entre ellos existen varias alternativas, sin embargo como ya se ha hecho mención se utiliza la distancia Euclidiana.

$$D_{st} = \sqrt{\sum_{i=1}^{64} (a_i - b_i)^2}$$

(6)

Donde D_{st} es la distancia Euclidiana que hay entre los 64 elementos de ambos descriptores. Realizando una serie de operaciones entre cada descriptor de la imagen A con cada una de la imagen B, se podrá decidir cuáles son correspondientes con mayor probabilidad, eligiendo siempre el que tenga la distancia Euclidiana menor.

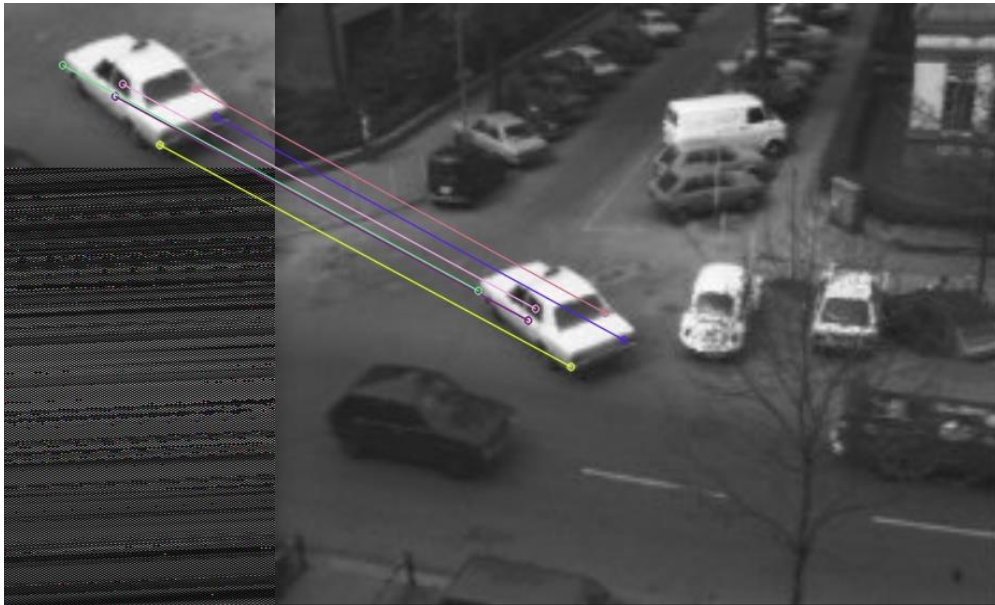


Figura 6: Correspondencia de características distancia euclidiana

El costo computacional total de las comparaciones es muy elevado si no se fijan restricciones. Por ejemplo, si se sabe con certeza que ciertos puntos de la imagen A no pueden aparecer de ningún modo en una zona determinada de la imagen B, se podrá descartar los descriptores de esa zona y así ahorrar el costo de las comparaciones. Por tanto, en función del tipo de aplicación que se quiera implementar y de la información que se tenga a priori, se podrá optimizar el código de una forma u otra.

10. RANSAC

Finalmente una vez que se tiene las correspondencias entre los puntos, tanto de la secuencia de video como del objeto de interés, se calcula la homografía utilizando el algoritmo de RANSAC. Sin embargo la cuestión de cómo adaptar una homografía o transformación afín a un conjunto de características coincidentes como es visible en la Figura 7.

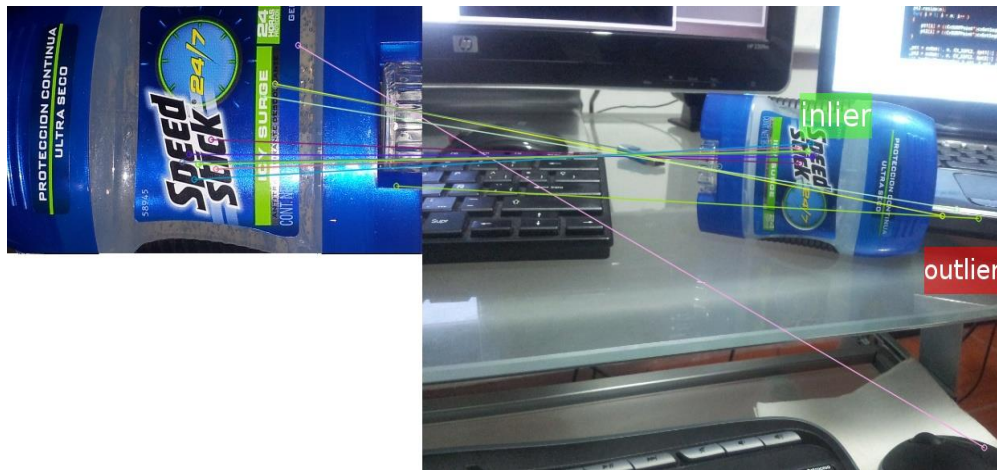


Figura 7: Valores atípicos en la correspondencia

Tiene algunos problemas: hay algunas correspondencias incorrectas. Ahora considerando una transformación afín la cual mapea un punto en otro punto, dando entonces T como:

$$T = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

(7)

Donde se tiene a su vez un conjunto de n correspondencias que son los puntos coincidentes en la Figura 7 como ejemplo. Ahora bien considerando entonces una correspondencia de un punto característico tenemos que:

$$[x_1, y_1] \rightarrow [x'_1, y'_1]$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ 1 \end{bmatrix}$$

$$ax_1 + by_1 + c = x'_1$$

$$dx_1 + ey_1 + f = y'_1$$

(8)

Entonces se tienen 2 ecuaciones, 6 incógnitas, de las cuales se necesitan al menos 3 correspondencias, las cuales pueden ajustar por mínimos cuadrados. Como se puede ver, es un sistema grande aunque es relativamente fácil de resolver. En otras palabras, encontrar una transformación afín T que se mapeen de la imagen uno lo más cerca posible de su correspondencia, una ejemplo es el que se muestra en la Figura 8.



Figura 8: Correspondencia de puntos característicos

Es entonces donde entra el algoritmo de RANSAC, para buscar la mejor hipótesis, que arroje parámetros que se apeguen a la transformación. Es decir

por medio de votación se descartan los puntos que no están dentro del conjunto de los puntos que son correspondientes a la imagen, es decir valores atípicos (outlier) que son visibles en la Figura 7. Resolviendo entonces el sistema de ecuaciones se resuelve la mejor T , la mejor transformación que adecue a los valores típicos (inlier) y a la deformación del objeto buscado, cuestión que se representa en la siguiente Figura 9.

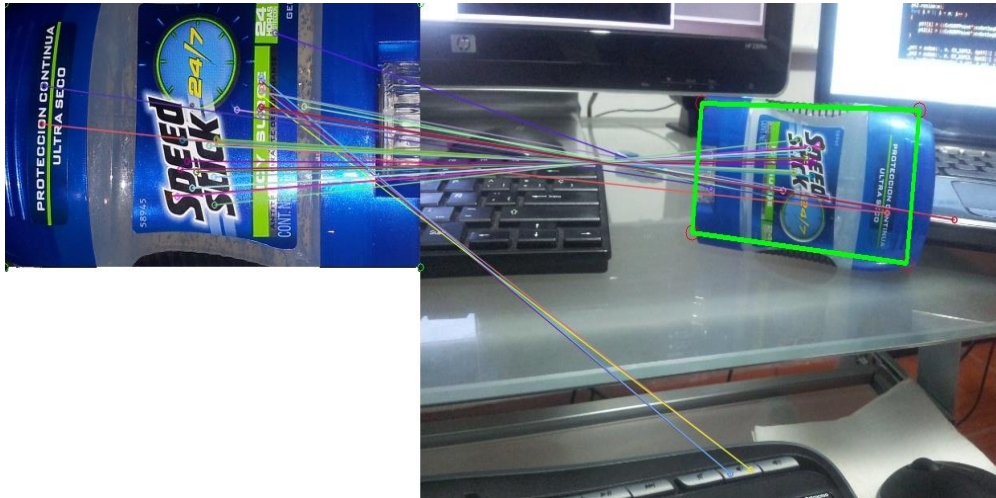


Figura 9: Correspondencia de un objeto mejores parámetros de T

11. Resultados

En la detección y seguimiento de objetos, el rendimiento y evaluación se reporta en base, a la verificación, identificación de un cierto conjunto de características, definidas por el objeto de interés, es decir, qué tan eficiente y acertada es la búsqueda del objeto deseado. No obstante la idea central de este proyecto es el seguimiento a partir de una sola muestra. Queda claro entonces que lo expuesto en los apartados anteriores, da como base el uso del algoritmo de SURF. Debido a los resultados arrojados en cuanto a su velocidad al generar los vectores de características y su robustez ante cambios de iluminación.

12. Correspondencia de muestra inicial

Con el fin de corroborar lo antes mencionado se hicieron pruebas con una sola muestra, obteniendo resultados notables, dado que la muestra inicial no correspondía a la forma que el objeto tenía en la imagen de búsqueda pero si los puntos característicos. Es decir, se podía detectar y seguir los puntos siempre y cuando siguieran siendo característicos. Por ende, los resultado son muy prometedores con una sola muestra, la siguiente Figura 10 muestra un ejemplo de lo antes mencionado.



Figura 10: Detección de objetos, imagen A logo de Ubuntu, imagen B discos collage de Ubuntu.

Donde es visible que la muestra no corresponde dado que se encuentra en sentido inverso y rotado algunos grados lo cual a pesar de estos cambios, el objeto es detectado y marcado, tratándose de apegar lo más posible a la muestra, imagen A, con su correspondiente en la imagen B.

Ahora bien, el fin del proyecto consiste en hacer el seguimiento en una secuencia de video, es entonces aquí donde las coordenadas del segundo conjunto de puntos característicos varia, en este caso los puntos generados de los marcos

de la secuencia de video. Luego entonces, se puede decir que se trata del mismo punto el cual a lo largo del seguimiento no deja de ser característico. Por ende, se puede hacer un seguimiento a partir de una sola muestra, como se puede ver en la siguiente Figura 11.



Figura 11: Detección de objetos en la secuencia de video Taxi de Hamburgo.

Se incluyen las 15 imágenes que componen el video o secuencia del Taxi de Hamburgo de izquierda a derecha numeradas. Dado que el trabajo, esta implementado en video y en tiempo real, y con esto se trata de ejemplificar el funcionamiento de la implementación. Por ende cabria resaltar que la demostración como tal valdría probarla en tiempo real.

13. Discusión

La formulación general del problema de la detección y seguimiento de un objeto, se puede plantear de la siguiente manera: dado una secuencia de vídeo, de una escena en particular, se debe determinar la posición (en la imagen) del objeto de interés, así como la trayectoria que éste genere en los cuadros subsecuentes. De igual forma, retomar la posición del objeto si el seguidor pierde el objeto de interés, validando el objeto a detectar con un conjunto de características que describan al objeto seguido. En cuanto a la parte de la detección, también se suele clasificar en dos problemas bien diferenciados: identificación y verificación, como se menciona en [Yilmaz et al 2006]. En los problemas de identificación, la entrada del sistema es la imagen de un objeto desconocido, marcando la identidad del mismo determinada a partir de un conjunto de características extraídas del seguimiento. Por otro lado, los problemas de verificación, el sistema necesita confirmar o rechazar la identidad pretendida de la imagen del objeto de entrada al proceso. Aunque existen muchas técnicas de seguimiento y detección de objetos, los enfoques que existen, han demostrado eficacia prometedora, pero las tareas de seguimiento y detección de objetos todavía son difíciles de completar satisfactoriamente, debido a las variaciones de iluminación, los contrastes del entorno donde se hace el seguimiento y la posición de los objetos con respecto a la cámara, así como las oclusiones que hacen que el seguidor pierda el objetivo, por ende, estos suelen ser algunos de los principales problemas de la detección y del seguimiento de objetos [Kalal et al., 2010]. El grado de complejidad aumenta en la medida que el entorno donde se implementa el seguimiento no es un ambiente controlado, por ejemplo: los lugares abiertos, al aire libre. Luego entonces, el problema que se aborda en este trabajo, radica en la posibilidad de generalizar el objeto de muestra inicial, al poder seleccionar cualquier objeto de interés de una secuencia de video, y a partir de esa muestra única, iniciar la detección y el seguimiento del objeto indicado. A diferencia de lo que existe en el estado del arte, no se contempla un entrenamiento previo, no existe un clasificador de base para poder hacer lo antes

mencionado. Otro problema importante en el que se centra este trabajo es: dado una única muestra inicial, poder seguir el objeto, a pesar de las deformaciones rígidas que se generen y los posibles cambios de iluminación, así como la posible pérdida del objeto de interés, por inclusiones de objetos diversos contenidos en la escena del video.

14. Conclusiones

La visión por computadora es un campo muy estudiado, pero a su vez muy profundo, con un sin número de posibilidades, de mejoras constantes a los diferentes algoritmos expuestos, a los diferentes métodos de búsquedas de objetos de interés y al universo que conlleva darle visión a un objeto inanimado cómo lo es una computadora o un robot.

En este trabajo, se ha implementado un sistema de seguimiento de objetos a partir de un solo ejemplo, dado por un click de ratón utilizando el algoritmo de SURF. Queda claro que el trabajo expuesto tiene un sin número de posibilidades, de mejoras, de aplicaciones, de desarrollo, en donde a partir del objeto que es seguido, se pueda aprender, a pesar de las deformaciones rígidas que del objeto emanen. Queda entendido que el algoritmo presentado como base de un sistema de seguimiento a tiempo real, pueden ser la base para proyectos de investigación posteriores. De igual forma queda claro la velocidad de cálculo del algoritmo de SURF es mucho más provechosa cuando de tiempo real se habla, pues el cálculo de los vectores de características son de mucho menor tamaño y su tiempo de computo es mucho menor que el de algunos otro algoritmos como lo es el algoritmo de SIFT. Ahora bien, cuando de un seguimiento en un video se habla la implementación del algoritmo de SURF para hacer seguimiento y detección, a pesar de tener un solo ejemplo, es bastante buena debido a que se pueda seguir lo indicado y retomarlo si el objeto de interés es perdido, y aunque existan deformaciones visibles del objeto, es capaz de adecuarse siempre y cuando los vectores de características sigan siendo característicos. Por ende se puede retomar el trabajo profundizando u

optimizando la búsqueda de correspondencias del objeto buscado. Por consiguiente, queda claro de igual forma, que es posible, basar el seguidor, no en un objeto específico, sino, generalizar lo que se quiere seguir, es decir, de una secuencia de video cualquiera, poder seleccionar cualquier objeto de interés que en ella exista.

Video descriptivo: <https://www.youtube.com/watch?v=RWBZM2Ps98A>

Referencias

Bay, H., Ess, A., Tuytelaars, T., y Gool, L. J. V. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3): páginas, 346–359.

Lindeberg, T. (1990). Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12: páginas, 234–254.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157

Kalal, Z., Mikolajczyk, K., y Matas, J. (2010). Face-1d: Tracking-learning-detection applied to faces. In *ICIP*, pages 3789–3792. IEEE.

Simard, P., Bottou, L., Haffner, P., y LeCun, Y. (1998). Boxlets: A fast convolutional algorithm for signal processing and neural networks. In Kearns, M. J., Solla, S. A., y Cohn, D. A., editors, *NIPS*, páginas, 571–577. The MIT Press.

Viola, P. A. y Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154

Yilmaz, A., Javed, O., y Shah, M. (2006a). Object tracking: A survey. *ACM Comput. Surv.*, 38(4).

Notas biográficas:

M.C. Jesús Eduardo Alcaraz Chávez. Actualmente se desempeña como profesor interino del Instituto Tecnológico de Morelia en el Departamento de Sistemas y Computación. Es Maestro en Ciencias de la Ingeniería Eléctrica orientación Sistemas y Computación, egresado del posgrado de la facultad de la Ingeniería Eléctrica de la UMSNH. Sus áreas de Interés son el procesamiento digital de imágenes y el procesamiento en tiempo real.

Dr. Félix Calderón Solorio. Actualmente se desempeña como profesor e investigador Titular C en la división de estudios de Postgrado de la facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo. Es doctor en ciencias computacionales por el Centro de Investigación en Matemáticas (CIMAT) ubicado en la ciudad de Guanajuato. Sus áreas de Interés son el procesamiento digital de imágenes. Ha sido miembro del Sistema nacional de Investigadores (SNI) y Perfil PROMEP. Actualmente es miembro del núcleo académico básico de la división de estudios de Postgrado de la facultad de Ingeniería Eléctrica de la UMSNH, cuyos programas de maestría y doctorado en Ingeniería Eléctrica, son reconocidos por el PNP del CONACyT.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.